Houghton, C. (2019). Calculating the Mutual Information between Two Spike Trains. *Neural Computation*, *31*(2), 330-343. https://doi.org/10.1162/neco_a_01155, https://doi.org/10.1162/neco_a_01155

Calculating the Mutual Information between Two Spike Trains

Conor Houghton
conor.houghton@bristol.ac.uk
Computational Neuroscience Unit, School of Computer Science, Electrical and Electronic Engineering, and Engineering Maths, University of Bristol, Bristol, Avon BS8 1UB, UK

It is difficult to estimate the mutual information between spike trains because established methods require more data than are usually available. Kozachenko-Leonenko estimators promise to solve this problem but include a smoothing parameter that must be set. We propose here that the smoothing parameter can be selected by maximizing the estimated unbiased mutual information. This is tested on fictive data and shown to work very well.

1 Introduction

Many problems in neuroscience are addressed by examining the relationship between the spiking output of two neurons. The best way to do this should be to calculate the mutual information between the two spike trains: the mutual information is a measure of how much information the spike trains share and is therefore an ideal way of quantifying the relationship between them. However, in practice, it has not been easy to use mutual information in this way because of the huge number of data required to estimate it. This is because in the established method for calculating information theory quantities for spike trains, the spike trains are converted into "words" by discretizing time; this produces a huge number of words, and so estimating their probabilities requires more electrophysiological data than it is typically practical to record.

In Houghton (2015) a Kozachenko-Leonenko estimator (Kozachenko & Leonenko, 1987; Victor, 2002; Kraskov, Stögbauer, & Grassberger, 2004; Tobin & Houghton, 2013) is presented for estimating the mutual information for random variables that take values on a metric space, that is, for data where there may be no coordinates but where it is possible to measure the distance between two data points. This method is referred to here as the density estimation method. The density estimation method is relevant to the study of neuronal data because there are many metrics on spike trains (Victor & Purpura, 1996; van Rossum, 2001; Aronov, Reich, Mechler, &

Neural Computation **31, 330–343** (2019)
doi:10.1162/neco_a_01155

Victor, 2003; Houghton & Sen, 2008; Houghton & Victor, 2010), which make the space of spike trains into a metric space.

This density estimation method for calculating the mutual information between spike trains relies on the choice of a smoothing parameter h. This letter describes an effective way to select this parameter and tests this method on fictive spike train data. It is found that the density estimation method produces a similar result to the more traditional binned method, but does so using considerably fewer spike train data.

The mutual information measures the dependence between two random variables U and V and is given by

$$I(U; V) = \left\langle \log_2 \frac{p_{U,V}(\mathbf{u}, \mathbf{v})}{p_U(\mathbf{u})p_V(\mathbf{v})} \right\rangle \tag{1.1}$$

where $\langle \ldots \rangle$ denotes the average with respect to the joint distribution $p_{U,V}(\mathbf{u}, \mathbf{v})$. $\mathbf{u}$ and $\mathbf{v}$ are values drawn from the U and V variables, respectively. In the application considered here, the aim is to estimate the mutual information between the activity of two neurons, so $\mathbf{u}$ and $\mathbf{v}$ are short intervals of spike train, one from each of the two neurons. In this letter, 45 ms intervals are used. In order to calculate the mutual information, the probability mass function $p_{U,V}(\mathbf{u}, \mathbf{v})$ needs to be estimated. The method for calculating mutual information described here is essentially a method of estimating the probability mass function.

The binned method for calculating the mutual information uses discretization. For a bin width δt, each spike train interval is converted into a word by binning the spikes and counting the number of spikes in each bin. The mutual information is calculated on the words rather than the spike trains with the probability of a given word estimated by counting how often it occurs in the data. The advantage of this method is that in the limit of vanishing δt and of an infinite number of data, the estimated mutual information approaches the true value. The disadvantage is that it approaches this true value very slowly. This is because of the huge number of words; for example, with 45 ms spike train intervals and $\delta t = 3$ ms, there are 2^{15} words and 2^{30}, that is, just over 1 billion, pairs of words corresponding to the spike train interval pairs. This situation can be improved with clever techniques (Treves & Panzeri, 1995; Nemenman, Bialek, & de Ruyter van Steveninck, 2004; Magri, Whittingstall, Singh, Logothetis, & Panzeri, 2009), but one basic limitation is that it considers each word individually. The power of a Kozachenko-Leonenko approach is that it exploits the proximity structure of a metric space, meaning the data points are considered in pairs.

Let $\mathcal{P} = \{(\mathbf{u}_1, \mathbf{v}_1), (\mathbf{u}_2, \mathbf{v}_2), \ldots, (\mathbf{u}_n, \mathbf{v}_n)\}$ denote a set of pairs of intervals from spike trains recorded during an experiment. These are modeled as being drawn from the probability distribution $p_{U,V}(\mathbf{u}, \mathbf{v})$. Following the formulation of the Kozachenko-Leonenko estimator given in Houghton (2015), this probability distribution is approximated by

$$p_{U,V}(\mathbf{u}_i, \mathbf{v}_i) \approx \frac{\#B}{\mathrm{vol}\,(B)}, \tag{1.2}$$

where B is a ball around the point $(\mathbf{u}_i, \mathbf{v}_i)$, $\#B$ is the number of points in the ball, and vol(V) is the volume of the ball. This approximation comes straight from the definition of the probability mass function,

$$\langle \#B \rangle = \int_B p_{U,V}(\mathbf{u}, \mathbf{v}) dV, \tag{1.3}$$

with the additional assumption that $p_{U,V}(\mathbf{u}, \mathbf{v})$ is approximately constant on B. The idea is that for every point, a ball of fixed volume is used to estimate the probability mass function at that point. The volume chosen for this ball is a smoothing parameter: the larger the volume, the more accurate $\#B$ estimates $\langle \#B \rangle$, but for larger volumes, the assumption that $p_{U,V}(\mathbf{u}, \mathbf{v})$ is approximately constant on B becomes less accurate.

The difficulty is how to calculate the volume of B. Because there are no useful coordinates for the space of spike trains, there is no *dxdydz*-style integration measure. However, a probability mass function does provide a volume measure on a space, and as described in detail in Houghton (2015), the marginalized distribution $p_U(\mathbf{u})p_V(\mathbf{v})$ can be used to provide a volume measure on the space of spike train pairs.

Though this seems an odd choice of volume measure, it gives a simple formula for the mutual information. For a point (u_i, v_i), consider the nearest h U-spike-train intervals to u_i,

$$C_U(\mathbf{u}_i, \mathbf{v}_i) = \{(\mathbf{u}_j, \mathbf{v}_j) : d(\mathbf{u}_j, \mathbf{u}_i) \text{ is one of the } h \text{ smallest } U\text{-distances}\}, \tag{1.4}$$

and the nearest h V-spike-train intervals to v_i,

$$C_V(\mathbf{u}_i, \mathbf{v}_i) = \{(\mathbf{u}_j, \mathbf{v}_j) : d(\mathbf{v}_i, \mathbf{v}_j) \text{ is one of the } h \text{ smallest } V\text{-distances}\}. \tag{1.5}$$

Now the ball around $(\mathbf{u}_i, \mathbf{v}_i)$ is defined as

$$C(\mathbf{u}_i, \mathbf{v}_i) = C_U(\mathbf{u}_i, \mathbf{v}_i) \cup C_V(\mathbf{u}_i, \mathbf{v}_i), \tag{1.6}$$

which has an estimated volume of h^2/n^2. Finally let $\#C(u_i, v_i)$ be the number of (u_j, v_j) points in $C(\mathbf{u}_i, \mathbf{v}_j)$:

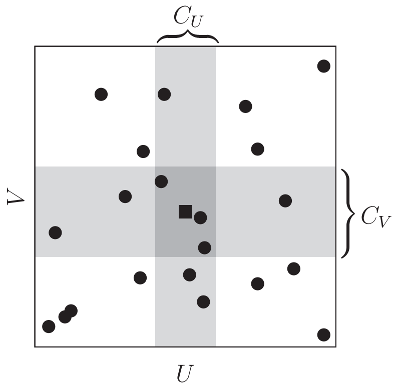

Figure 1: The calculation of $I_{\mathrm{KL}}(\mathcal{P}; h)$. The U-space corresponds to the horizontal direction and the V-space to the vertical. Of course, this is a cartoon; these spaces are not one-dimensional, and they do not even have a defined dimension. The points in $\mathcal{P}$ are marked as filled circles and a square; the square is the one whose contribution to $I_{\mathrm{KL}}(\mathcal{P}; h)$ is being calculated. The gray bars represent $C_U(\blacksquare)$ and $C_V(\blacksquare)$ with $h = 7$; that is, each of the gray rectangles contains seven points. $\#C(\blacksquare) = 4$ since it counts the points in the intersection, that is, the region with darker shading.

$$\#C(u_i, v_i) = \#[C_U(\mathbf{u}_i, \mathbf{v}_i) \cap C_V(\mathbf{u}_i, \mathbf{v}_i)]. \tag{1.7}$$

With this notation, the Kozachenko-Leonenko approximation for the mutual information is

$$I(U; V) \approx I_{\mathrm{KL}}(\mathcal{P}; h) = \frac{1}{n} \sum_{i=1}^{n} \log_2 \frac{n\#[C(u_i, v_i)]}{h^2}. \tag{1.8}$$

This quantity is straightforward to calculate. For each point $(\mathbf{u}_i, \mathbf{v}_i)$, the set $C_U(\mathbf{u}_i, \mathbf{v}_i)$ contains $(\mathbf{u}_i, \mathbf{v}_i)$ itself, and the nearest $h - 1$ points to $(\mathbf{u}_i, \mathbf{v}_i)$ when $\mathbf{u}_i$ is compared to the $\mathbf{u}_j$ in other $(\mathbf{u}_j, \mathbf{v}_j)$ pairs. Similarly, the set $C_V(\mathbf{u}_i, \mathbf{v}_i)$ contains the nearest $h - 1$ points when $\mathbf{v}_i$ is compared to $\mathbf{v}_j$. $\#C(\mathbf{u}_i, \mathbf{v}_i)$ is the size of the intersection. This is illustrated in Figure 1.

It is instructive to consider what happens if the two distributions are independent. In this case, it is possible to calculate the probability that $\#C(\mathbf{u}_i, \mathbf{v}_i) = r$ for different possible values r; it is a sort of urn problem. Choosing the $h - 1$ points in $C_U(\mathbf{u}_i, \mathbf{v}_i)$ that are not $(\mathbf{u}_i, \mathbf{v}_i)$ itself is like randomly selecting $h - 1$ points out of $n - 1$, and calculating r is to ask how many are in $C_V(\mathbf{u}_i, \mathbf{v}_i)$. This gives

$$\text{prob}\,(\#C(\mathbf{u}_i, \mathbf{v}_i) = r) = \frac{\binom{h-1}{r-1}\binom{n-h}{h-r}}{\binom{n-1}{h-1}}, \tag{1.9}$$

so

$$I_0(n, h) = \sum_{r=1}^{h} \text{prob}\,(\#C(\mathbf{u}_i, \mathbf{v}_i) = r) \log_2 \frac{nr}{h^2} \tag{1.10}$$

is the estimated mutual information when the two distributions are independent. This is an upward bias in the estimate of the mutual information; an upward bias is a common feature of estimators of mutual information. In this case, the bias is because B will not always contain exactly $\langle \#B \rangle$ points. One advantage of the Kozachenko-Leonenko approach is that I_0 gives an explicit formula for the bias, and it depends only on the smoothing parameter h and the number of pairs, n.

Obviously, as h approaches n, this bias approaches zero, but otherwise, it is positive, and as a bias, it can be removed from the estimate of the mutual information:

$$I(U; V) \approx \tilde{I}(\mathcal{P}; h) = I_{\text{KL}}(\mathcal{P}; h) - I_0(n, h). \tag{1.11}$$

Recall that there are two competing approximations used in deriving the estimate. For small h, the counting estimates for the number of points in a ball and for the volume of the balls are noisy. For large h, the estimate of the probability mass function is too smooth. The first of these approximations is the cause of the bias described by $I_0(n, h)$. Conversely, $I_0(n, h)$ is not affected by the smoothing bias. This suggests that the best approximation is found by maximizing $\tilde{I}(\mathcal{P}; h)$ over h:

$$I(U; V) \approx \tilde{I}(\mathcal{P}) = \max_h \tilde{I}(\mathcal{P}; h). \tag{1.12}$$

It is demonstrated here that this works very well.

2 Methods

2.1 Data. The algorithm is run on fictive data generated using two leaky integrate-and-fire neurons with shared input. The two neurons satisfy

$$\tau_m \frac{dv_i}{dt} = E_l - v_i + I_i, \tag{2.1}$$

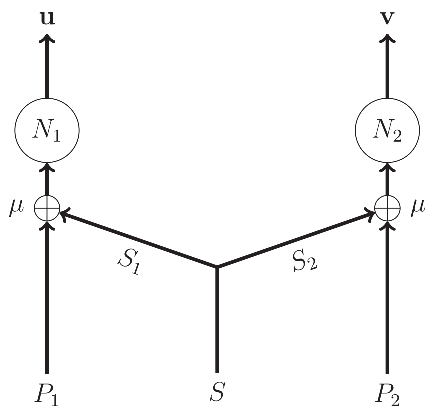

Figure 2: The fictive data. This shows the network used to produce the fictive data for testing the density estimation formula for mutual information. N_1 and N_2 are both leaky integrate-and-fire neurons, producing the spike trains **u** and **v**, respectively. Their input is a weighted average of P_i and S_i. S is a shared input with $S_1 = S$ and $S_2 = \bar{S} - S$.

where $i = 1, 2$ labels the two neurons, $\tau_m = 12$ ms and $E_l = -70$ mV. If $v_i > V_t$, where $V_t = -70$ mV, a spike is recorded and v_i is reset to E_l. There is a refractory period of $\tau_r = 2$ ms. I_i is an input. It is a voltage rather than a current because it has absorbed the membrane resistance:

$$I_i = (1 - \mu)P_i + \mu S_i, \tag{2.2}$$

where P_i is an input particular to the i neuron, whereas S_i is an input based on a shared input S:

$$\begin{aligned} S_1 &= S, \\ S_2 &= \bar{S} - S, \end{aligned} \tag{2.3}$$

with $\bar{S} = 30$ mV and the parameter μ specifying the amount of common input. The inputs P_i and S are both piecewise constant, with each having a fixed value for a period chosen independently from an exponential distribution with mean $\tau_c = 30$ ms. The value for each interval is chosen uniformly from $[0, \bar{S}]$. This network is illustrated in Figure 2.

This method of producing fictive data is not perfect in the sense that the temporal correlation is different for different values of μ and the firing rate varies from 32 Hz at $\mu = 0$ and $\mu = 1$ to 27 Hz at $\mu = 0.5$. However, the aim is to test the spike train pairs with different values of mutual information. As will be seen, this method succeeds in doing that.

2.2 The Distance between Spike Trains. For the density estimation information calculation, the distance between individual spike trains is calculated using the van Rossum metric (van Rossum, 2001). This calculates the distance between two spike trains $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ and $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ as

$$d(\mathbf{u}, \mathbf{v}) = \sum_{i,j} e^{-|u_i - u_j|/\tau} + \sum_{i,j} e^{-|v_i - v_j|/\tau} - 2\sum_{i,j} e^{-|-u_i - v_i|/\tau}, \tag{2.4}$$

where τ is a timescale that can be thought of as expressing the precision of spike times in neuronal coding. A value around $\tau = 15$ ms is often used. In the formula for mutual information, the metric is used to order points by proximity, so it might be expected that the values of estimated mutual information would not depend in a detailed way on the distance values or the choice of metric. In fact, it will be seen here that in these data, this holds true: the results are not sensitive to the value of τ.

2.3 Calculating the Information. The pairs of spike trains produced by the network model are chopped up into 45 ms intervals. To calculate the mutual information using the binned method, these intervals are discretised with $\delta t = 3$ ms giving 15-letter words. The frequency for each word pair is estimated from the data using the obvious empirical estimate:

$$p_{U,V}(\mathbf{u}, \mathbf{v}) \approx \frac{\text{Occurrences of } (\mathbf{u}, \mathbf{v})}{\text{Total number of samples}}. \tag{2.5}$$

This is then used to calculate the mutual information directly. The bias is removed by also calculating the mutual information for shuffled data. In this case, this means shuffling the pairing between the spike train intervals (Nirenberg, Carcieri, Jacobs, & Latham, 2001; Montemurro, Senatore, & Panzeri, 2007; Panzeri, Senatore, Montemurro, & Petersen, 2007; Magri et al., 2009).

To calculate the mutual information using the density estimation method, the distance matrix for the U-spike trains and the V-spike trains were calculated using the efficient implementation of the van Rossum metric described in Houghton and Kreuz (2012). The optimal value of h was found using a golden mean search (Kiefer, 1953).

3 Results

In Figure 3 the value of the mutual information for 45 ms intervals of spike train is calculated for different values of $\mu \in [0, 1]$ using both the binned and density estimation approaches. For the density estimation approach, 200 s of data are used; for the binned method, 25,000 s of data are used to

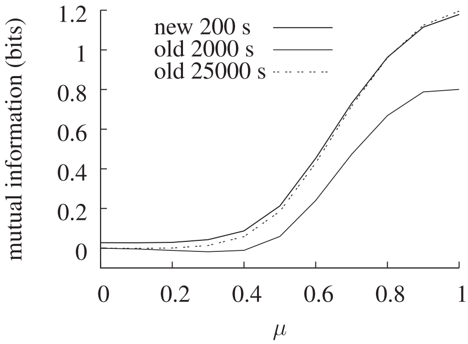

Figure 3: Estimates of the mutual information for different values of μ. The mutual information between pairs of 45 ms fragments of spike train is estimated using the density estimation (marked "new") and the binned (marked "old") approaches. In the case of the density estimation approach, 200 s of spike train are used. In the binned approach, 2000 s and 25,000 s are used. In all cases, the graph shows the average of 100 trials. In the density estimation approach, $\tau = 15$ ms; in the binned approach, 3 ms bins are used.

establish the ground truth and a smaller value of 2000 s to illustrate the number of data needed to estimate the mutual information.

Using 2000 s of data, the binned method gives a very poor estimate of the mutual information for most values of μ; the density estimation method is much closer to the value estimated using 25,000 s of data. The binned method is better for values of $\mu < 0.4$ when the amount of mutual information is very low. Presumably this is because the noise in the estimate is more significant and the maximization over h leads to an overestimate.

Figures 4A and 4B show the convergence of the density estimation and binned methods; Figure 4C uses a log scale to exhibit both on the same graph. These graphs show that the density estimation method uses considerably fewer data than the binned method. It is clear from these graphs that the estimators approach their asymptotic values in an orderly way. This means one approach (described in Treves & Panzeri, 1995; Strong, Koberle, de Ruyter van Steveninck, & Bialek, 1998; Panzeri et al., 2007) to improving the estimate using the binned method is to fit the graph to a curve such as

$$I(a, b, c) = a + \frac{b}{\sqrt{T}} + \frac{c}{T\sqrt{T}}, \tag{3.1}$$

where T is the length of spike train used. For the simulated data being examined here, this works quite well, concentrating on $\mu = 0.7$. As an example, fixing a, b, and c using the first 2000 s gives an estimate of 0.6613 for

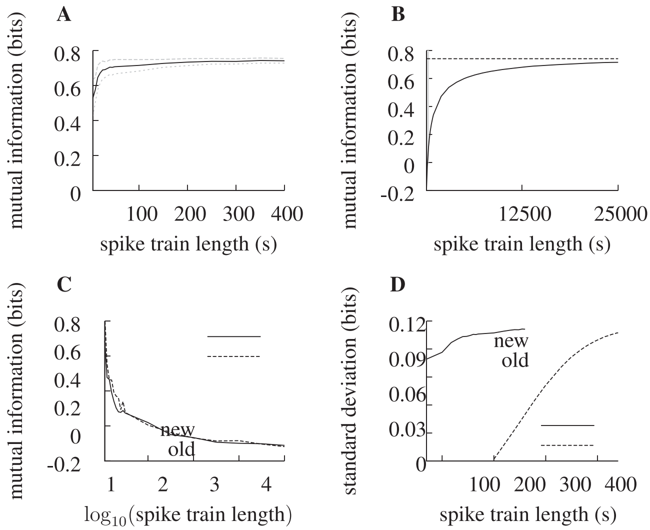

Figure 4: Performance of the formula for calculating mutual information. All of these graphs concern pairs of spike trains where $\mu = 0.7$ and the mutual information is being calculated between 45 ms fragments. Panels A and B compare the density estimation (A) and binned (B) approaches to calculating mutual information as longer and longer spike trains are used. In panel B, the very thin gray rectangle along the vertical axis marks out the area shown in panel A and the horizontal line gives the value estimated by the density estimation approach using 400 s spike trains. These two graphs are shown again in panel C, where a log-scale is used for spike train length. In both panels A and B, the plots are of the mean over 100 trials. In panel A, the dotted lines show 1 standard deviation from the mean. For panel B, the standard deviation is vanishingly small because so many data are used in this approach. In panel D, the standard deviations using the density estimation and binned approaches are compared. They are roughly similar, though in the binned approach, the mean is very different from the value estimated using more data.

$T = 25{,}000$ s, compared to an actual value of 0.7156. The value given by the density estimation method using 400 s of data is 0.7412. The binned method gives even larger values if even larger numbers of data are used; for this value of μ, 1,000,000 s of data give 0.768413. Extrapolating the binned method from 200 s of data does not work; it gives an estimate of 0.1975. Figure 4D compares the standard deviation for the two methods; they are roughly the same.

The robustness of the density estimation approach to mutual information is examined in Figure 5. In Figures 5A and 5B, the lengths of the short intervals used to calculate the mutual information are changed. Figure 5A

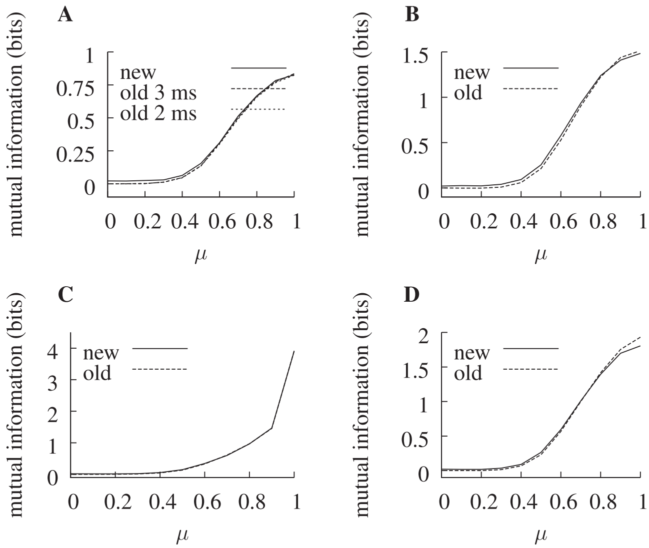

Figure 5: Testing the robustness of the density estimation approach. In panels A and B, the length of the spike train intervals is changed. In panel A, they are shortened to 30 ms; in panel B, they are increased to 60 ms. In each case, for the binned approach (marked "old"), the intervals are discretized into 3 ms letters. In panel A, the binned approach with 2 ms letters is also plotted. In panel C, an alternative model is used to produce the simulated data: the shared input is the same, so the input is $S_i = S$ for both values of i. In panel D, $\bar{S} = 35$ mV is used to generate the simulated data. This increases the firing rate so that it varies from 44 Hz at $\mu = 0$ and $\mu = 1$ to 39 Hz at $\mu = 0.5$. For the binned approach, 25,000 s of spike trains are used in each case. For the density estimation approach (marked "new"), 200 s of spike trains are used in panels A and C and 500 s in panels B and D. When the intervals contain on average more spikes, the density estimation approach appears to require more data. In panel D, the difference between the two approaches is more noticeable for μ near one than in other graphs.

also plots the binned estimate with a different letter length. In Figure 5C, a different stimulus is used, whereas for all the other simulations, the shared input is shared with $S_1 = S$ and $S_2 = \bar{S} - S$. In this figure, $S_i = S$ for both values of i. Finally, in Figure 5D, an input with a higher firing rate is used. The density estimator performs well, but there is some indication that the number of data required, though modest compared to the binned approach, increase as the number of spikes increases.

The sensitivity of the estimate to the choice of metric is examined in Figure 6. In Figure 6A the van Rossum metric is replaced by the Victor-Purpura metric (Victor & Purpura, 1996). This was the first metric proposed

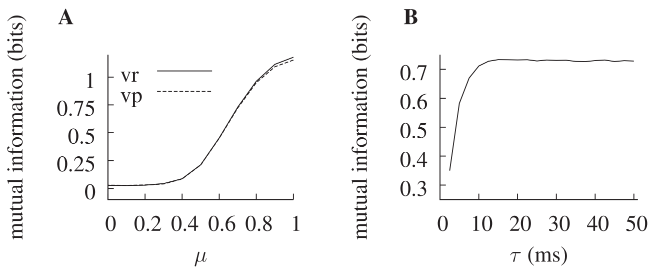

Figure 6: The effect of the metric. (A) The density estimation approach is applied using the Victor-Purpura metric instead of the van Rossum metric. This comparison uses the same simulated data as Figure 3. The estimated information using the van Rossum metric with $\tau = 15$ ms is marked "vr." The Victor-Purpura metric has a "cost" parameter q. Like τ in the van Rossum metric, it expresses the precision of spike times in coding. Here, $q = 2/\tau$, and the corresponding estimate is marked "vp." The two estimates are nearly identical. (B) The value of τ used in the metric in the density estimation approach is varied; generally the estimate does not depend sensitively on the value of τ.

for spike trains and rivals the van Rossum metric in measures of how well metric distances capture information coding in spike trains (Houghton & Victor, 2010). Furthermore, it is a non-Euclidean metric (Aronov & Victor, 2004). The van Rossum metric works by embedding the space of spike trains into the infinite-dimensional space of functions. In this sense, the van Rossum metric is Euclidean. Replacing it with the Victor Purpura metric demonstrates that this Euclidean property is not required for the density estimation approach to work. In Figure 6B small values of τ show poor performance; for small values of τ, the metric distance between two spike trains is very dependent on noise, which jitters spike times to a degree that is significant when compared to τ. For values of τ that are similar or larger than the size of the interval, the relative distances between different pairs of spike train do not change as τ varies. The behavior of the smoothing parameter is explored in Figure 7. Figure 7A shows an example of how the estimated mutual information depends on h, and Figure 7B graphs the change in the optimal value of h as the length of the spike trains changes.

4 Discussion

This letter describes a method for choosing the smoothing parameter for a Kozachenko-Leonenko estimator of the mutual information and tests it on fictive spike train data. It is seen that the density estimation method is very effective in estimating the mutual information using much smaller numbers of data than required for the discretization-based approach.

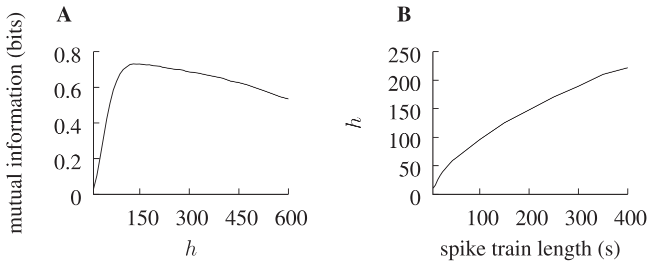

Figure 7: The smoothing parameter h. In both graphs, the mutual information is estimated for $\mu = 0.7$. (A) The estimated mutual information is plotted against h using 200 s of spike train. (B) The optimal value of h, that, is the h maximizing the estimated information, is plotted as a function of the spike train length.

I hope that this approach will prove useful in estimating mutual information for real data. Obviously, using the approach in practice will require a choice of the interval length and spike train metric. Typically, as with the binned approach, the interval length should be chosen to reflect the timescale of interest for the cells being studied. This might, for example, reflect the membrane constants of the cells and the correlation length of their stimuli. The hope is that the density estimation method will allow longer interval lengths to be used than was possible with the binned approach. If the van Rossum metric is used, the parameter τ needs to be set. Ideally, this value should maximize the estimated mutual information; often $\tau = 15$ ms is used for van Rossum metrics. There are also measures of spike train similarity Kreuz and coworkers (Kreuz, Chicharro, Houghton, Andrzejak, & Mormann, 2012; Kreuz, Haas, Morelli, Abarbanel, & Politi, 2007) that adapt to the time-local spike rate and do not have a parameter like the τ in the van Rossum metric or the q in the Victor-Purpura metric. These would avoid the need to fix the metric parameters.

The density estimation approach is more computationally demanding than the binned approach. The whole matrix of distances between pairs of interval pairs must be calculated, and for each interval pair, the nearest h other pairs need to be found.

In the density estimation method, the information is estimated from the matrix of distance values. This means that any information-carrying features of the spike trains that are not captured by the metric will be lost in the estimate of mutual information. Spike train metrics are often evaluated using transmitted information (Victor & Purpura, 1996; Houghton & Victor, 2010). They are, in this sense, designed to capture the information-carrying features. However, mutual information estimated from a distance matrix must underestimate the true value. In the example considered here, this underestimate appears to be small; there may be some indication that the underestimate increases as the number of spikes increases. This might be more significant for real data, where there may be information-carrying motifs in spike trains. There are none in the simulated data.

The Kozachenko-Leonenko estimator is a powerful approach to calculating mutual information based on the proximity structure of the data. It is often more efficient than estimators that do not incorporate this structure. It has also been shown in Houghton (2015) that it does not require useful coordinates for the spaces the random variables take their values on. Obviously this is the case when the data of interest are spike train data, but there are likely to be manifold other applications to other data types, including other applications involving neuroscience data, such as calculating the mutual information between spiking responses and a continuous stimulus space, as previously considered in Panzeri, Treves, Schultz, and Rolls (1999).

Acknowledgments

Thanks to the James S. McDonnell Foundation for support through a Scholar Award in Cognition (JSMF #220020239; https://www.jsmf.org/). This work was carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol (http://www.bris.ac.uk/acrc/).

References

Aronov, D., Reich, D. S., Mechler, F., & Victor, J. D. (2003). Neural coding of spatial phase in V1 of the macaque monkey. *Journal of Neurophysiology*, *89*(6), 3304–3327.

Aronov, D., & Victor, J. D. (2004). Non-Euclidean properties of spike train metric spaces. *Physical Review E*, *69*(6), 061905.

Houghton, C. (2015). Calculating mutual information for spike trains and other data with distances but no coordinates. *Royal Society Open Science*, *2*(5), 140391.

Houghton, C., & Kreuz, T. (2012). On the efficient calculation of van Rossum distances. *Network: Computation in Neural Systems*, *23*(1–2), 48–58.

Houghton, C., & Sen, K. (2008). A new multineuron spike train metric. *Neural Computation*, *20*(6), 1495–1511.

Houghton, C., & Victor, J. (2010). Measuring representational distances—the spike-train metrics approach. In N. Kriegeskorte & G. Kreiman (Eds.), *Visual population codes: Toward a common multivariate framework for cell recording and functional imaging* (pp. 391–416). Cambridge, MA: MIT Press.

Kiefer, J. (1953). Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society*, *4*(3), 502–506.

Kozachenko, L., & Leonenko, N. N. (1987). Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, *23*(2), 9–16.

Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical Review E*, *69*(6), 066138.

Kreuz, T., Chicharro, D., Houghton, C., Andrzejak, R. G., & Mormann, F. (2012). Monitoring spike train synchrony. *Journal of Neurophysiology*, *109*(5), 1457–1472.

Kreuz, T., Haas, J. S., Morelli, A., Abarbanel, H. D., & Politi, A. (2007). Measuring spike train synchrony. *Journal of Neuroscience Methods*, *165*(1), 151–161.

Magri, C., Whittingstall, K., Singh, V., Logothetis, N. K., & Panzeri, S. (2009). A toolbox for the fast information analysis of multiple-site LFP, EEG and spike train recordings. *BMC Neuroscience*, *10*(1), 81.

Montemurro, M. A., Senatore, R., & Panzeri, S. (2007). Tight data-robust bounds to mutual information combining shuffling and model selection techniques. *Neural Computation*, *19*(11), 2913–2957.

Nemenman, I., Bialek, W., & de Ruyter van Steveninck, R. (2004). Entropy and information in neural spike trains: Progress on the sampling problem. *Physical Review E*, *69*(5), 056111.

Nirenberg, S., Carcieri, S. M., Jacobs, A. L., & Latham, P. E. (2001). Retinal ganglion cells act largely as independent encoders. *Nature*, *411*(6838), 698.

Panzeri, S., Senatore, R., Montemurro, M. A., & Petersen, R. S. (2007). Correcting for the sampling bias problem in spike train information measures. *Journal of Neurophysiology*, *98*(3), 1064–1072.

Panzeri, S., Treves, A., Schultz, S., & Rolls, E. T. (1999). On decoding the responses of a population of neurons from short time windows. *Neural Computation*, *11*(7), 1553–1577.

Strong, S. P., Koberle, R., de Ruyter van Steveninck, R., & Bialek, W. (1998). Entropy and information in neural spike trains. *Physical Review Letters*, *80*(1), 197.

Tobin, R. J., & Houghton, C. J. (2013). A kernel-based calculation of information on a metric space. *Entropy*, *15*(10), 4540–4552.

Treves, A., & Panzeri, S. (1995). The upward bias in measures of information derived from limited data samples. *Neural Computation*, *7*(2), 399–407.

van Rossum, M. (2001). A novel spike distance. *Neural Computation*, *13*(4), 751–763.

Victor, J. D. (2002). Binless strategies for estimation of information from neural data. *Physical Review E*, *66*(5), 051903.

Victor, J. D., & Purpura, K. P. (1996). Nature and precision of temporal coding in visual cortex: A metric-space analysis. *Journal of Neurophysiology*, *76*(2), 1310–1326.

Received July 10, 2018; accepted September 22, 2018.